ONF Core Model

Introduction to models, guidelines and tooling from

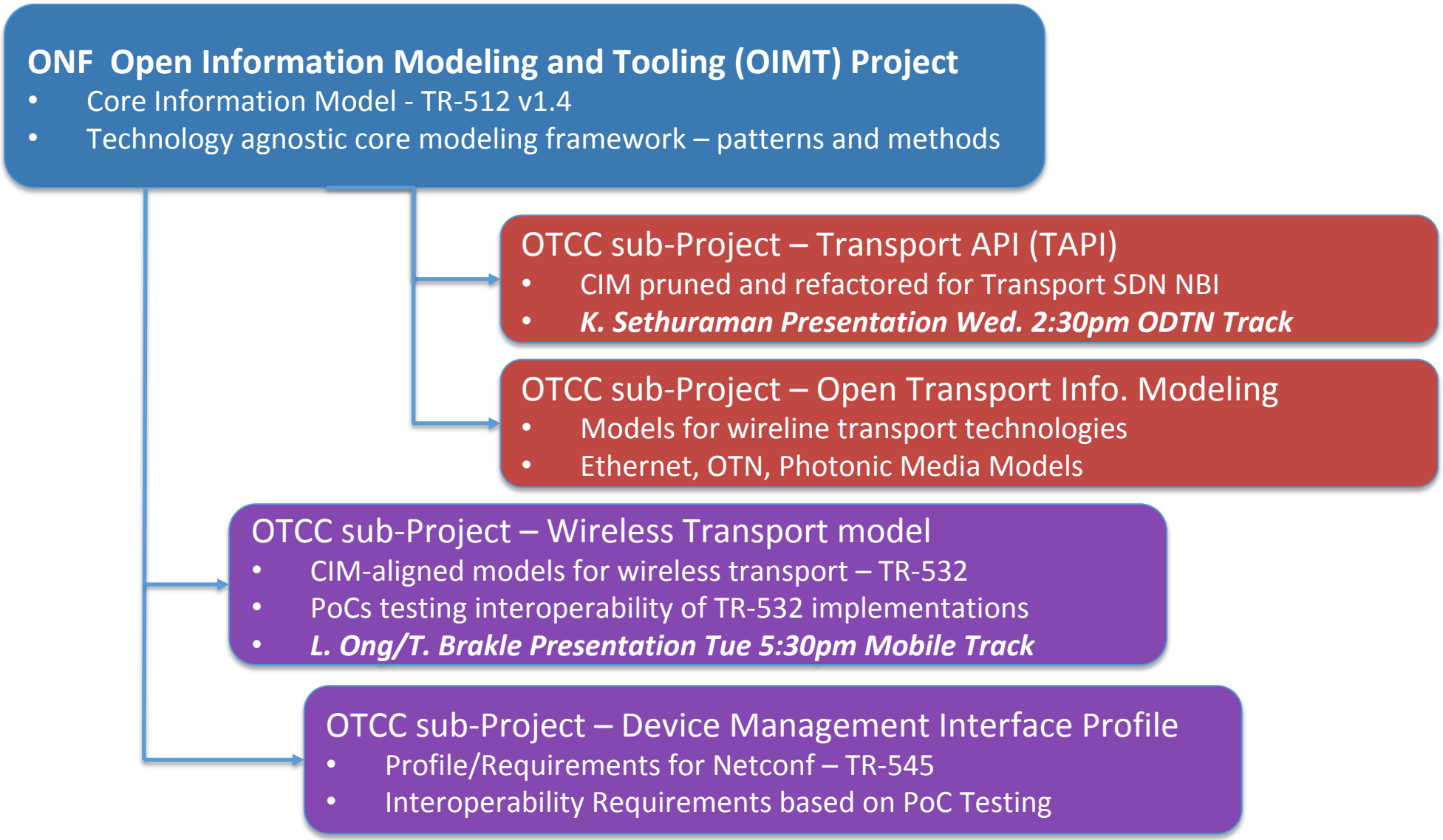
ONF Open Information Model & Tooling project

Nigel Davis (Ciena)

Kam Lam (FiberHome)

20181205

ONF API Modeling



Modelling SDN

The Core model provides a standardized implementation neutral representation of things and the relationship between those things in the SDN problem space

- Network functions. Model focus:
  - Virtualized termination/forwarding in any network
- Physical Equipment supporting the network. Model focus:
  - Field Replaceable Units (FRUs), non-FRUs, strands etc.
- Control functions supporting the network. Model focus:
  - Representation of functions related to closure of control loops
  - Presentation of views of the resources for the purpose of control
- Processing functionality supporting/using the network. Model focus:
  - Any abstract function
- Resource/System/Scheme specifications. Model focus:
  - Constraints, rules and specs for the of the overall systems
- Software supporting the control
  - Files, Installed Software, Containers, VMs,

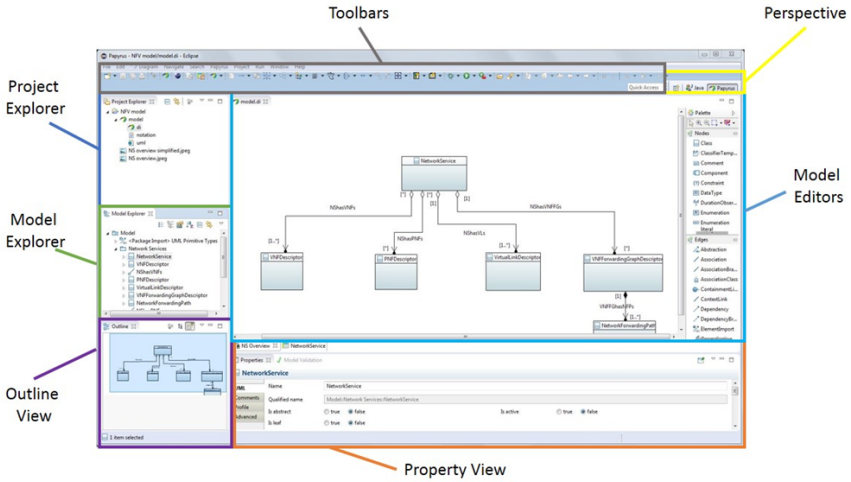
Most recent focus has been on Analogue Guided Media networks, using photonic networks as the key application.

TR-512.A.4 provides the explanation of the use of the Core Model for photonic networks.

This work has been used extensively by OTCC and Facebook TIP

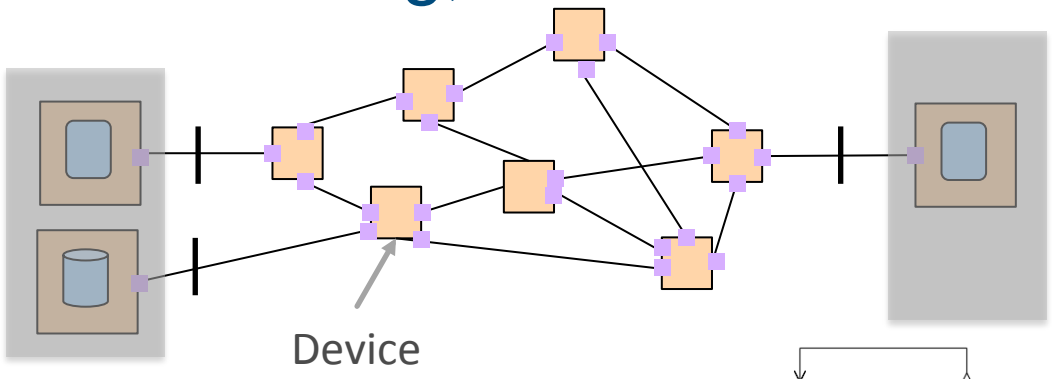
Model to create a common language

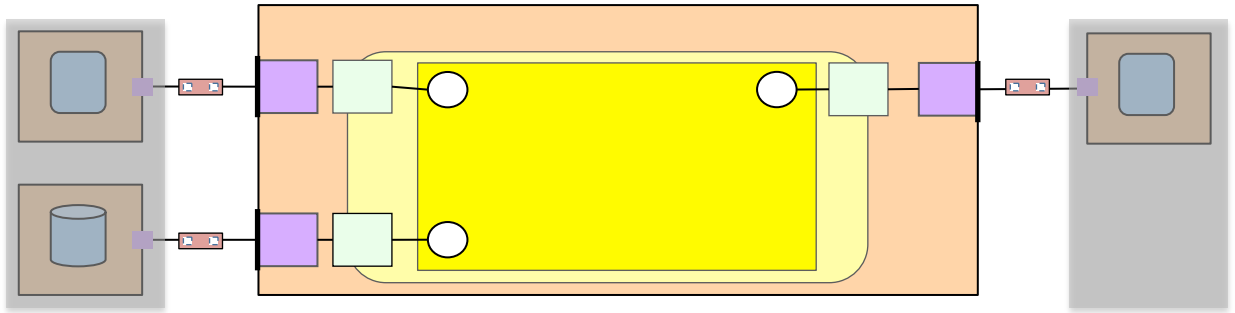
- Goals:
  - A well defined widely applicable representation of the semantics of managed network functionality that is lightweight, has a modular architecture and is technology/technique agnostic
  - Reduce the formation of overlapping inconsistent implementations which hinder overall progress
- Approach:
  - Leverage industry best-practices, patterns and tools to close the model to implementation round trip loop
  - Use Agile modelling methodology to construct a formal model using Papyrus UML
    - A graphical modeling language highlights underlying patterns
    - The environment provides a framework for:
      - Development of understanding about control of networks
      - Capturing a representation of the understanding
      - Maintaining growing insight
    - Promote Core Model use/extension
- Use:
  - Derivation of Interface/database models using generators to generate consistent artefacts in JSON, Yang etc.



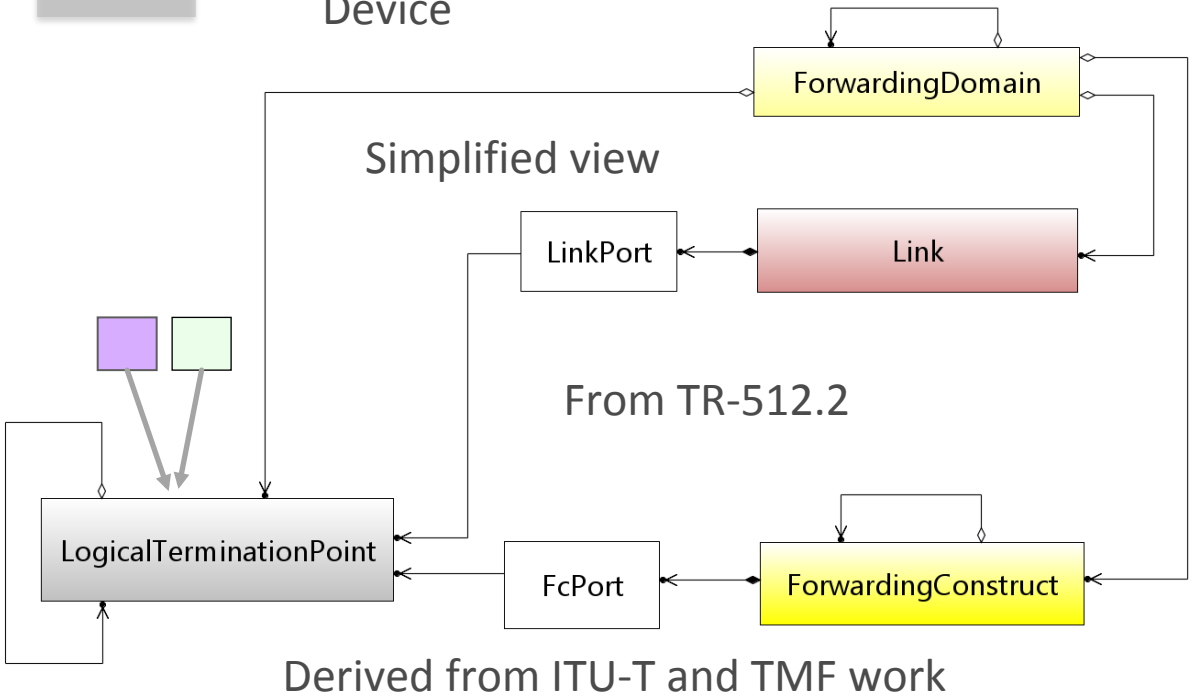
Canonical network model (virtualized/functional): Forwarding, Termination and Topology

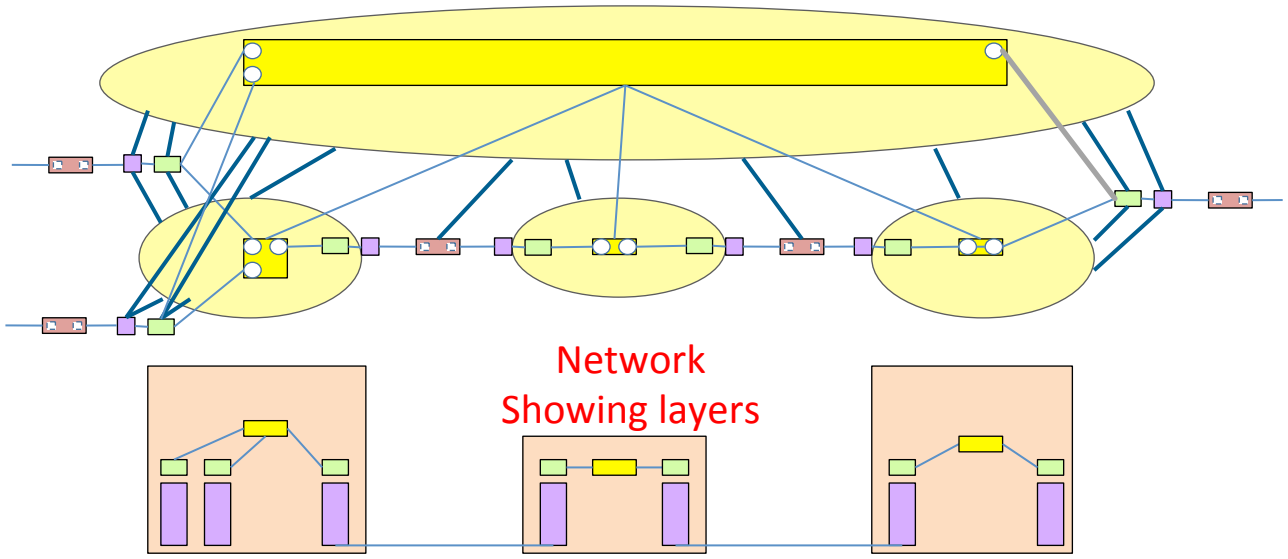
See slide notes





Network Device OR Opaque Network

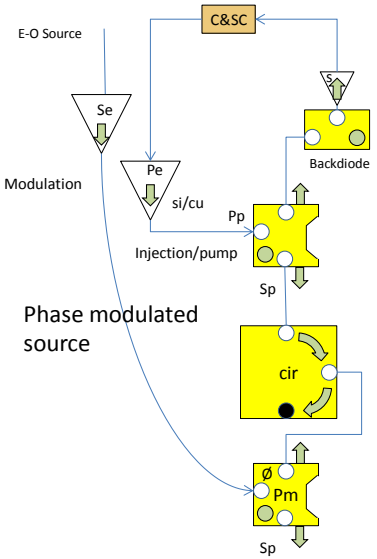




Network Showing layers

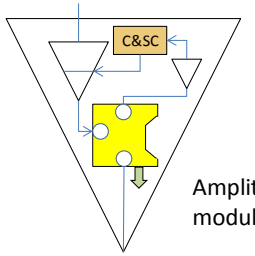
Model for any networking, for any network technology, with any degree of virtualization, at any scale, at any abstraction and in any interrelated view.

Photonic network analysis and modeling



Phase modulated source

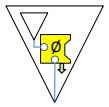
Detailed functions



Amplitude modulated source

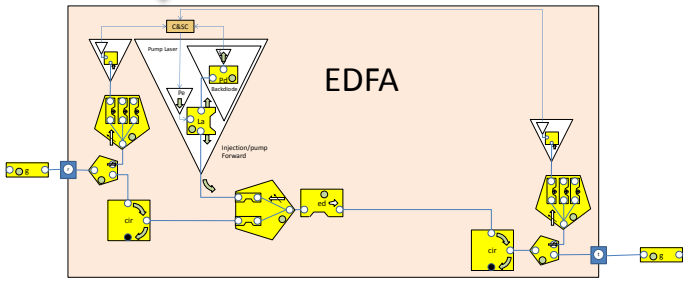
Detailed functions

Photonic Sources



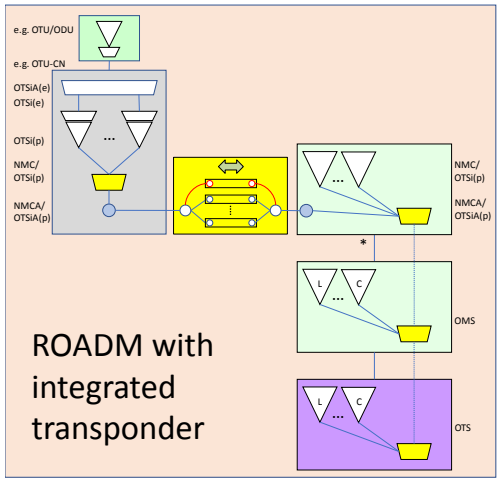
Abstracted symbol

Modelling of fundamental functions



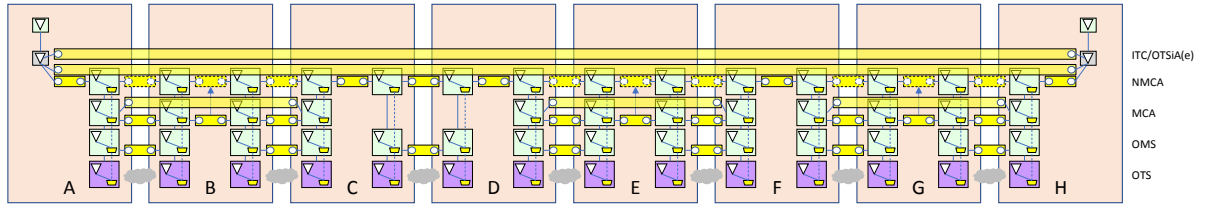
Modelling of Aggregated functions

Modelling of Devices



ROADM with integrated transponder

Modelling of Networks



ROADM and Amplifier Chain

Key (Core Model Entities)

- ForwardingConstruct (FC)
- LogicalTerminationPoint (LTP)
  - Intermediate
  - Floating
  - Tied to physical
- Device etc. (ConstraintDomain)
- Termination (in an LTP)
- Adapter (in an LTP)
- C&SC — ConfigurationAndSwitchController

From TR-512.A.4

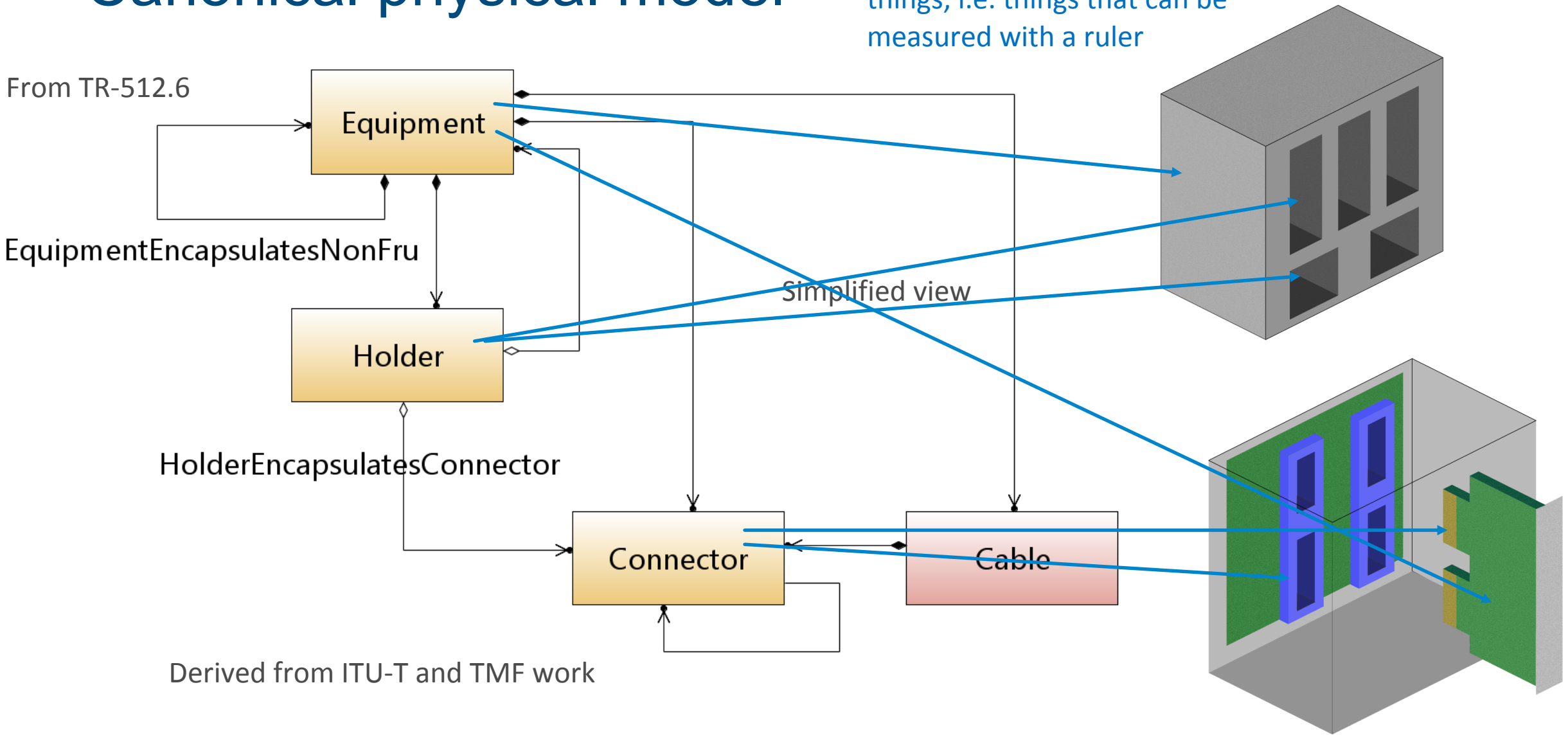
Using the model entities to represent photonics at all scales gives a consistent model regardless of the degree of aggregation etc.

Canonical physical model

Represents truly physical things, i.e. things that can be measured with a ruler

See slide notes

From TR-512.6

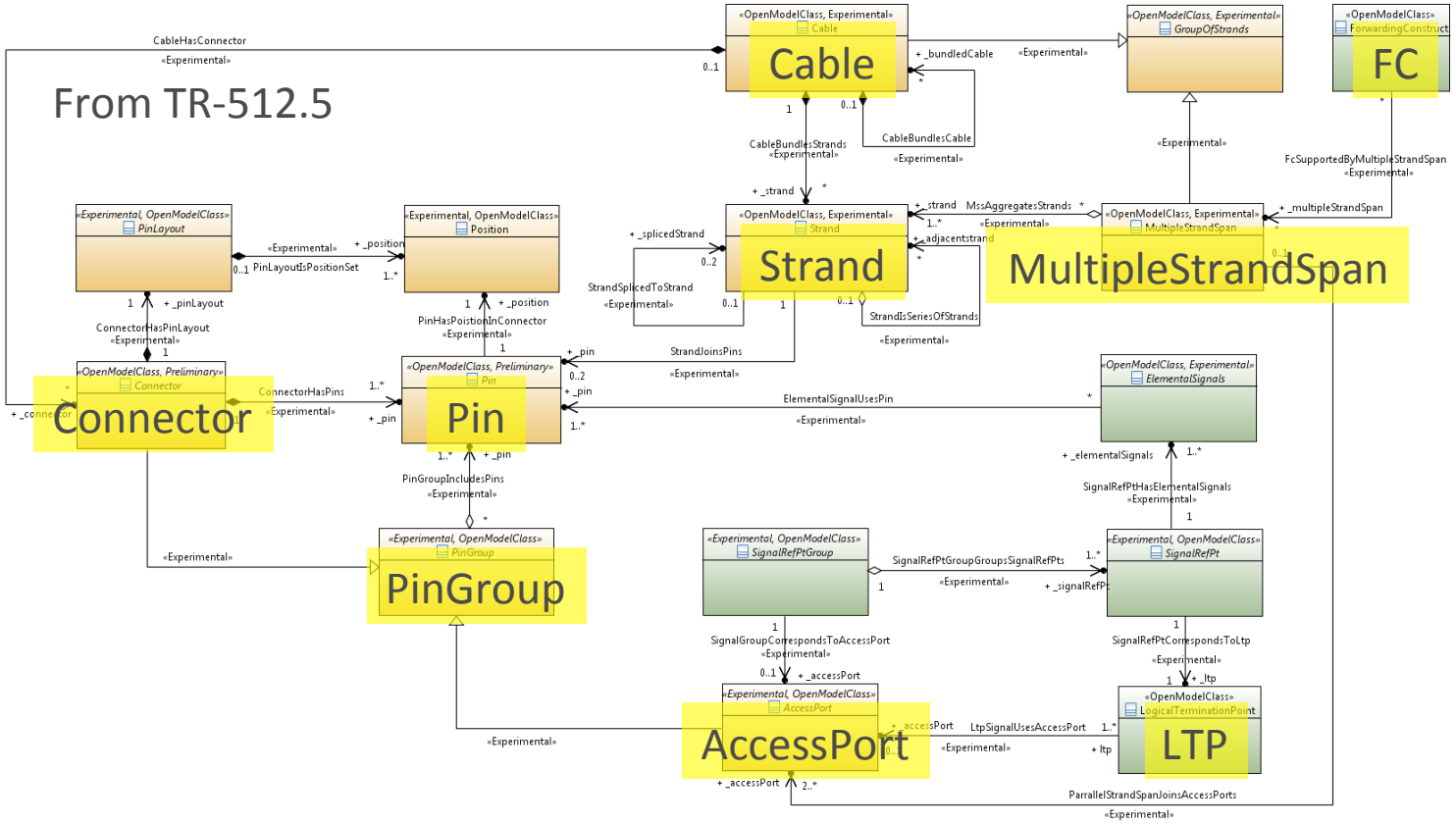


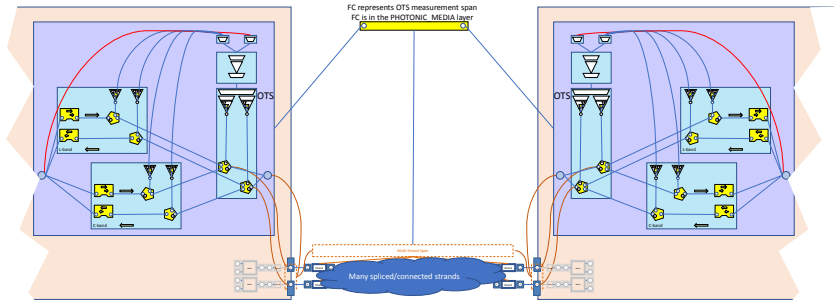
Derived from ITU-T and TMF work

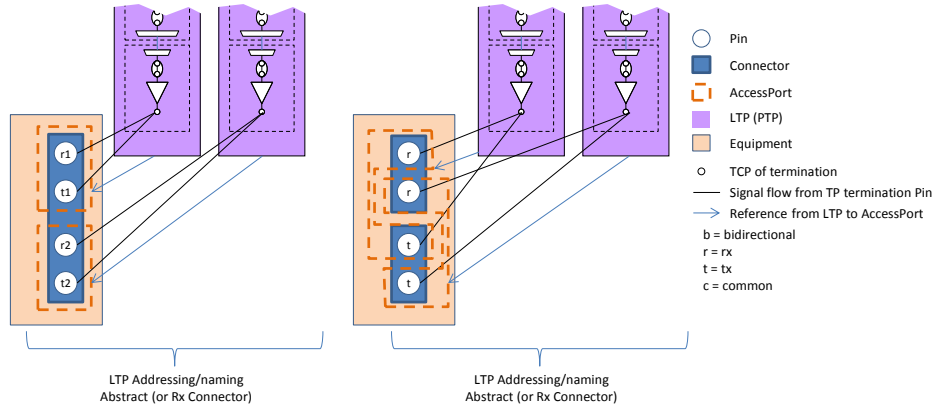
Model for any physical components that are rack/cabinet/shelf based or stand-alone in a data center or telco environment

Connector, Cable, AccessPort and MultipleStrandSpan

From TR-512.5







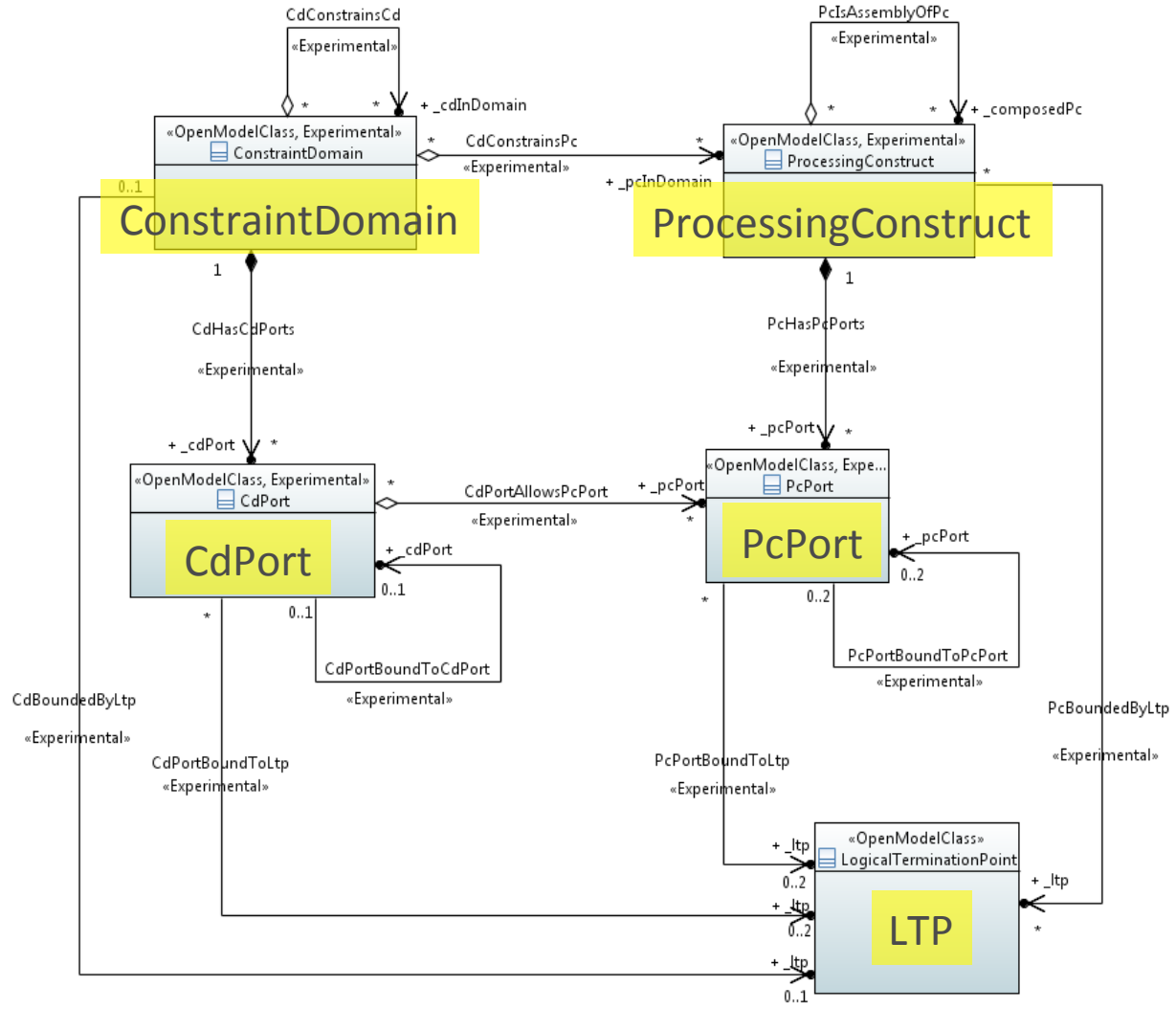
Strong separation of concerns of physical from functional yields a model that deals consistently with a wide range of physical structure

Model of generalized processing and constraint

See slide notes

From TR-512.11

Follows the Component-Port and Component-System patterns



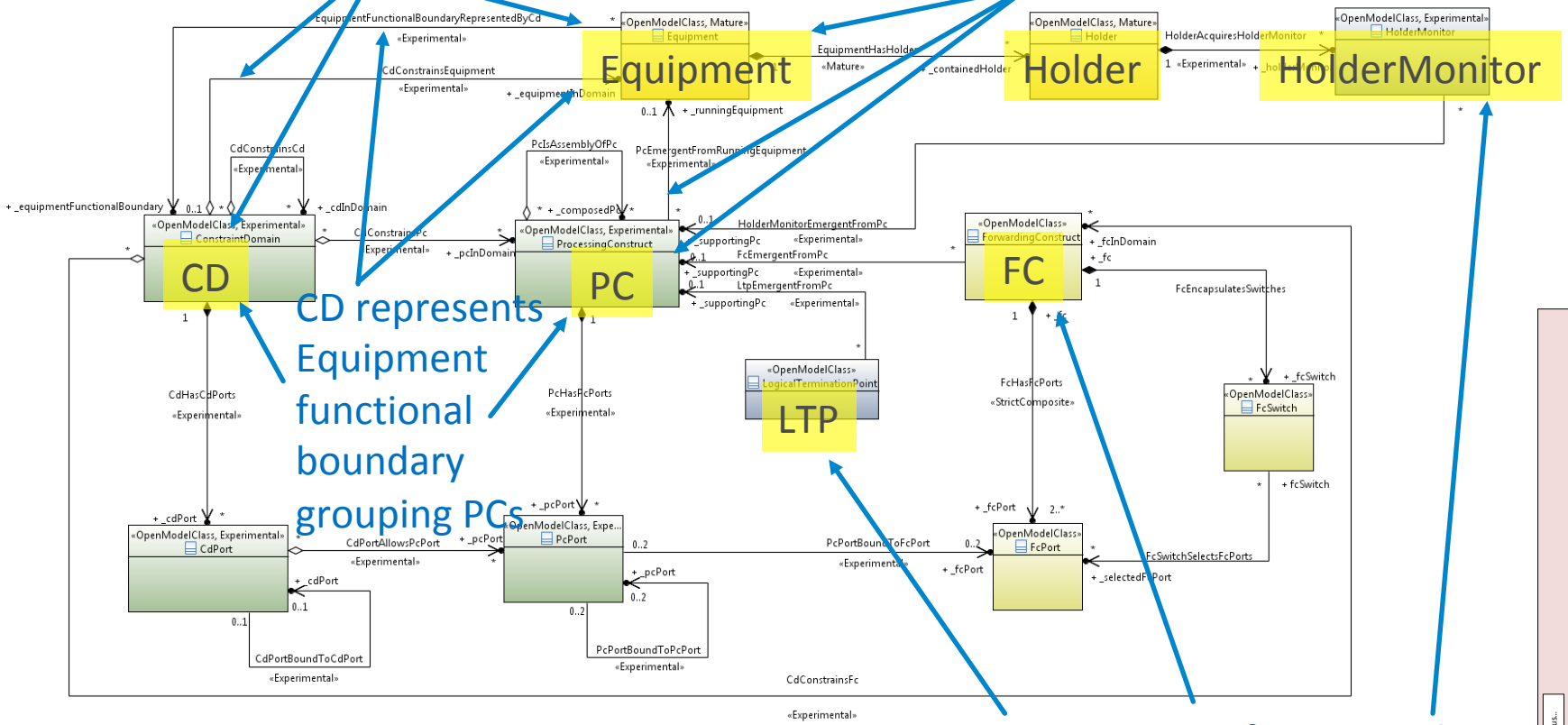
Model for any arbitrary functions/constraint, views of abstractions of functions/constraints, interconnection of functions to networking.

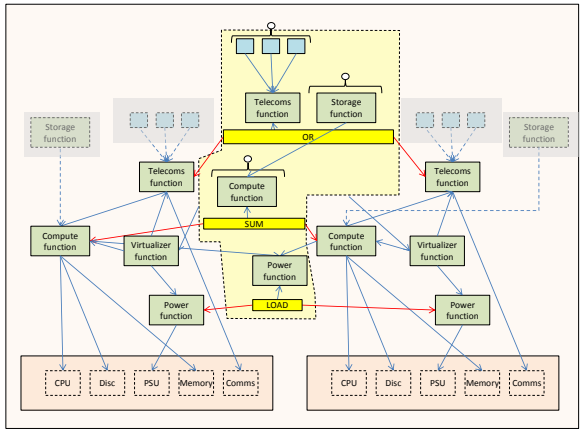
Association between physical and functional model

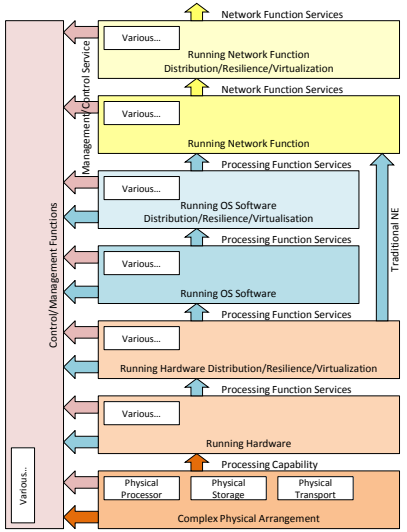
See slide notes

CD provides boundary around equipment to represent NE/ Device/Controller etc

PC emergent from running equipment







- Each block (Function, Software, Hardware, Physical and Control/Management) may be a single component or may be many components interacting to form one or more systems
- Each component may expose detail as if many components where the exposed components are interacting to form one or more systems
- Each set of services may be subdivided and may be complex reflecting the exposed components and their encapsulated functions
- Running OS Software is generalized to any OS including the negligible OS of a FPGA
- Distribution and resilience may not be present or may be present for only some of the services (management/control may also have distribution/resilience (not shown)
- A network function may be built directly on the Processing Function Services of the Running Hardware (i.e. is essentially a hardware realization) or some combination of running hardware and OS software
- Processing capability is a generalized term that could also apply to a parallel compute engine and a scattered set of dedicated engines such as any assembly of logic gates and buffers
- Each block is controlled/managed by a function that itself is built on Processing Capability and Processing Function Services
- The following figures use this colour scheme and often omit Management/Control

Derived from ITU-T and TMF work

From TR-512.6

Basis for model for the understanding of the physical realization of functional things (regardless of how “virtual” they are).

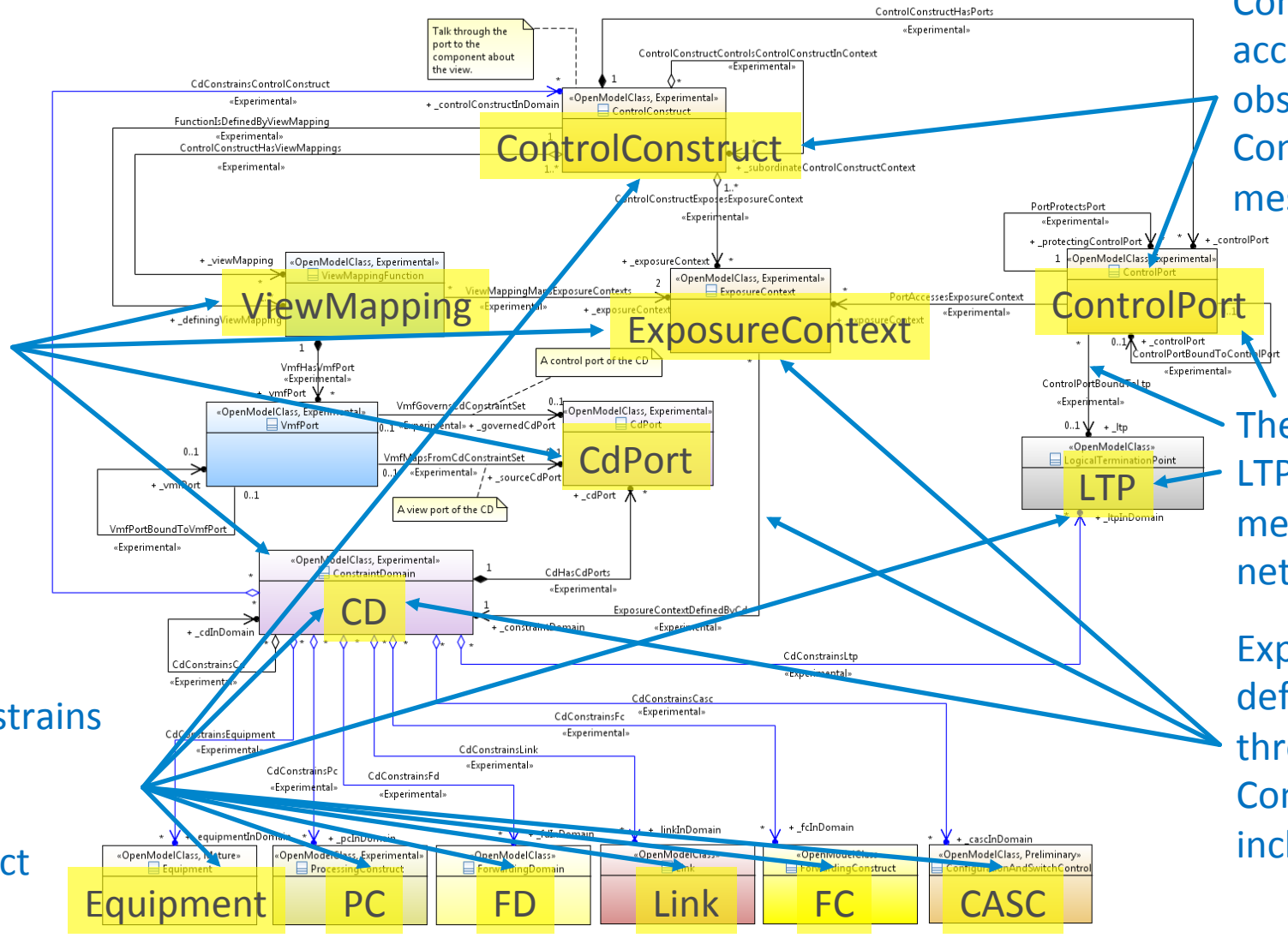
Model of control access and view

See slide notes

From TR-512.8

ControlConstruct provides access via its ports to enable observation/control. ControlPort provides the messaging/signalling

ViewMapping provides the relationship views defined by CD that are offered through different ExposureContexts



The ControlPort is bound to LTP allowing control messages to enter the network

ExposureContext provides definition of what is exposed through a ControlPort using ConstraintDonain (and hence including ControlConstruct)

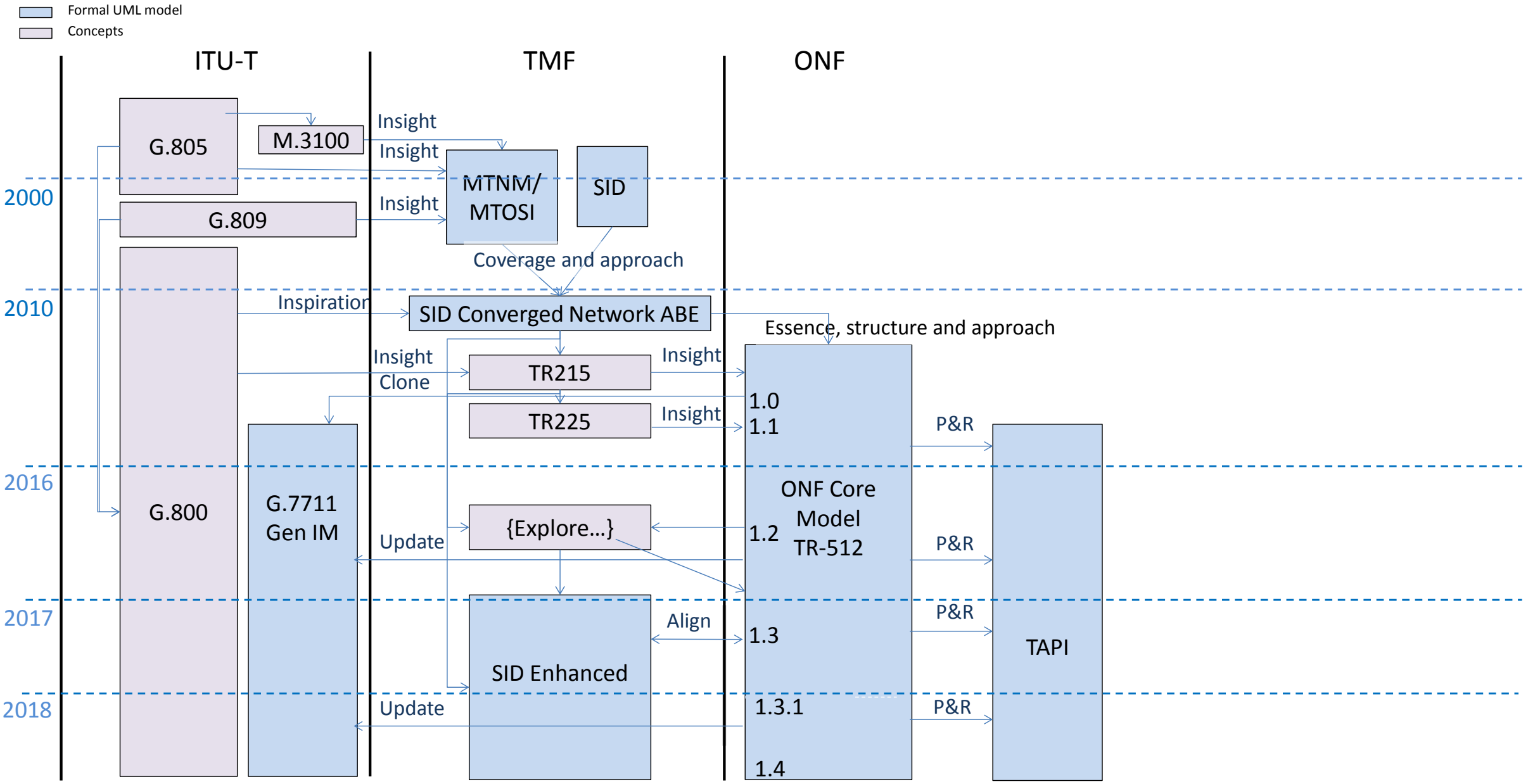
CD groups/constrains any structures including ControlConstruct

Model for the control functions in the network, for control of those control functions, and modelling of Control itself, for control of the Controller.

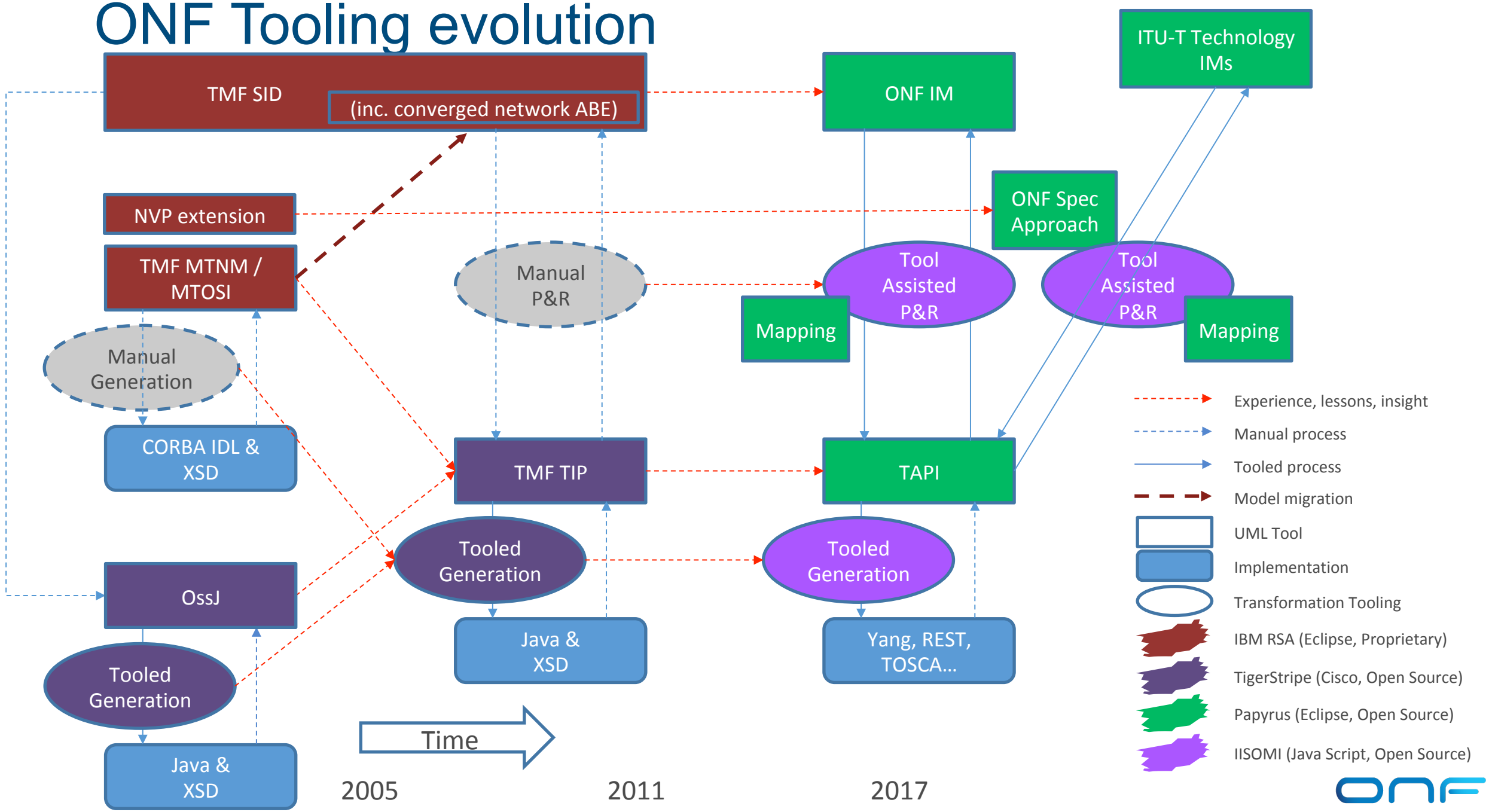
Other further key coverage in TR-512 V1.4

- Replacement of the NE and the SDN Controller with a uniform model of ControlConstruct, ExposureContext and ConstraintDomain [TR-512.8]
- Model of Control of Switching etc. using the ConfigurationAndSwitchController (CASCC) [TR-512.5]
- Component-System pattern and Component-Port pattern [TR-512.A.2]
- Augmentation via decoration defined by machine readable specification [TR-512.7]
- Modelling of Software [TR-512.12]

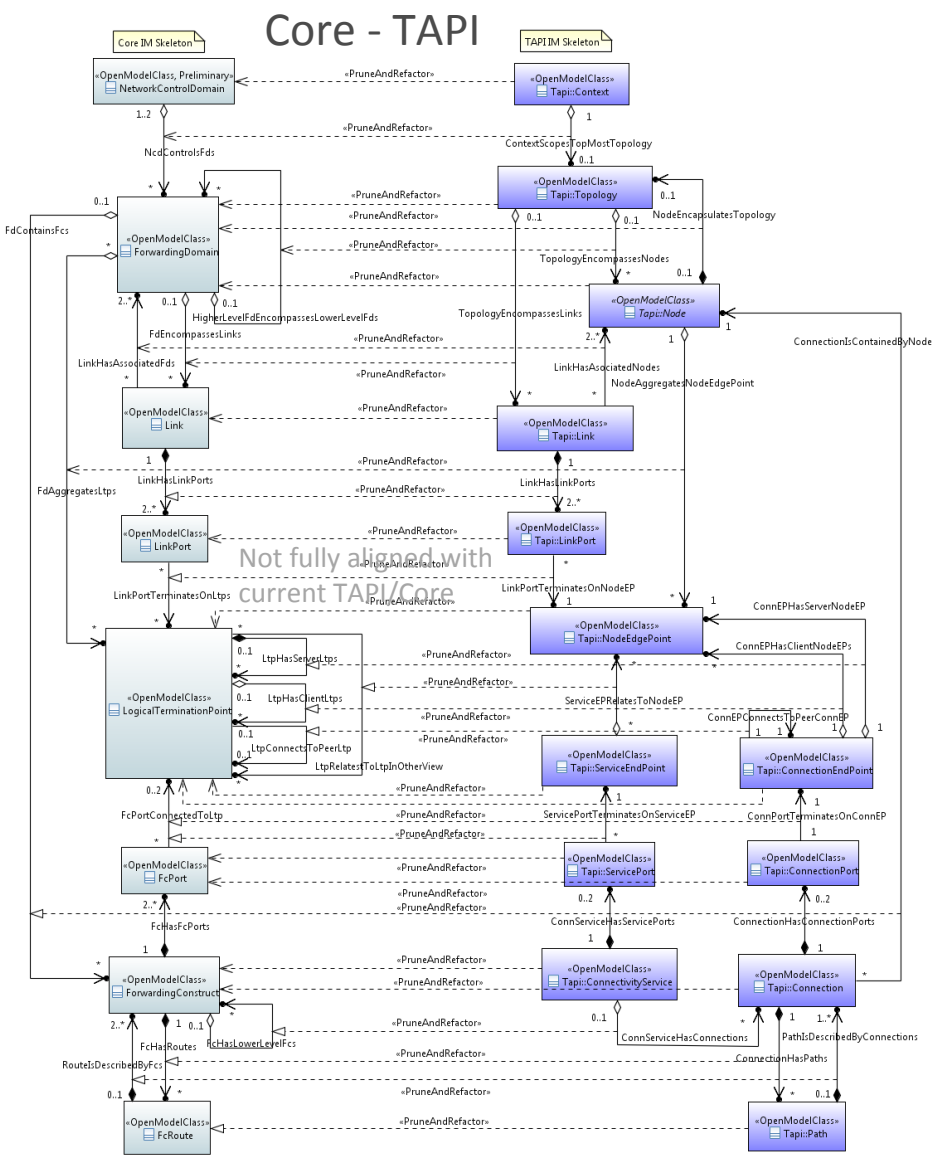
Information Model evolution



ONF Tooling evolution



ONF Core model to TAPI and WT models



- WT model is pruned from the Core model
  - The classes are unchanged in name and scope
  - Some classes/associations/attributes have been removed
- TAPI model is pruned and refactored from the Core model
  - The figure sketches the more complex relationships
  - The model is pruned in a similar way to WT
  - Some classes from the core are split
    - E.g., FD becomes Node and Topology in TAPI
  - Some classes are cloned and narrowed
    - E.g., FC becomes Connection and ConnectivityService
  - Renaming of classes is carried out to assist adoption in the application context

Ongoing work update

- Continue work with other bodies
  - MEF
  - OASIS-TOSCA
  - ONAP
  - ITU-T (publish TR-512 as G.7711)
  - TMF (in the process of adopting TR-512.2, TR-512.4 and TR-512.7 in place of the TMF SID 5LR model)
  - TIP
  - ITU-T/IEEE/MEF on Ethernet, OAM etc.
- TR-512 v1.5
  - Adopting the OAM model from TAPI
  - Enhancing the resilience model
  - Refining the Entity lifecycle
- TR-512 v 2.0
  - Consider adding ports to the LTP
  - Consider restructuring the model packaging

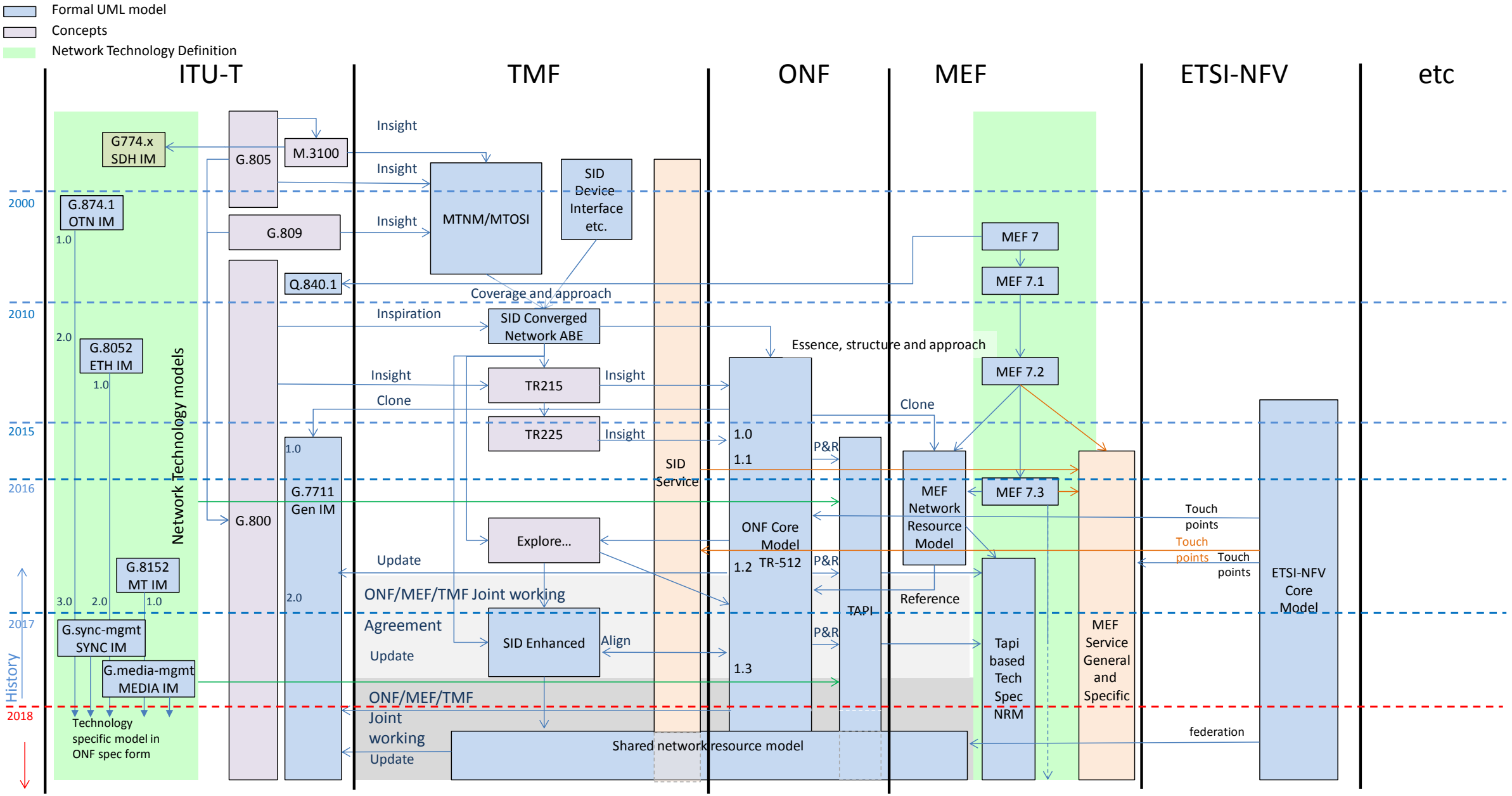
Links….

- Core model: TR-512 V1.4 (November 2018)
  - V1.4 can be found at https://3vf60mmveq1g8vzn48q2o71a-wpengine.netdna-ssl.com/wp-content/uploads/2018/12/TR-512_v1.4_OnfCoreIm-info.zip
  - See also https://www.opennetworking.org/software-defined-standards/models-apis/
- TAPI: V2.1.0
  - https://github.com/OpenNetworkingFoundation/TAPI
- Microwave model
  - TR-532 documents the model (see https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-532-Microwave-Information-Model-V1.pdf)
- UML Modeling Guidelines (IISOMI 514)
  - Last published version → v1.3 info (at https://www.opennetworking.org/software-defined-standards/models-apis/)
  - Latest working draft → Draft v1.3.01 (11/2018) (at https://wiki.opennetworking.org/display/OIMT/Infrastructure+Sub-team+Guidelines)
- UML Profiles and Style Sheets
  - Github repository: UmlProfiles (formal)
    - OpenModelProfile, v0.2.13
    - OpenInterfaceModelProfile, v0.0.8
    - ProfileLifecycleProfile, v0.0.4
    - Style sheet for class diagrams
  - and https://github.com/bzeuner/EAGLE-Open-Model-Profile-and-Tools/tree/ToolChain/UmlProfiles (latest)
- Papyrus Guidelines (IISOMI 515)
  - Last published version → v1.3 info (at https://www.opennetworking.org/software-defined-standards/models-apis/)
  - Latest working draft → Draft v1.3.01 (11/2018) (at https://wiki.opennetworking.org/display/OIMT/Infrastructure+Sub-team+Guidelines)
- Papyrus https://www.eclipse.org/papyrus/
- UML to YANG Mapping Guidelines (IISOMI 531)
  - Last published version → v1.1 info (at https://www.opennetworking.org/software-defined-standards/models-apis/)
  - Latest working draft → Draft v1.2 (11/2018) (at https://wiki.opennetworking.org/display/OIMT/UML+-+YANG+Guidelines)
- UML to YANG Mapping Tool
  - Github repository: https://github.com/OpenNetworkingFoundation/EagleUmlYang

Questions?

Thank you ☺

Information Model evolution



Abstract

- This presentation will explain the ONF Core model TR-512, a standardized implementation neutral representation of things and the relationship between those things in the SDN problem space. The presentation will show how the model has been developed through many years of practical experience.
- The presentation will work through the things explaining their relevance and application including: • Specific network functions focussing on virtualized termination and forwarding designed to represent any network • Generalized processing functionality designed to represent any abstract function • Real physical equipment specifically representing FRUs, non-FRUs, strands etc • Control functions related to presentation of views, closure of loops etc • System/scheme specifications representing the structures of systems
- The presentation will explain how the model can be applied at any level of abstraction to any control/ orchestration/automation layer and will work through the approach used to construct view model using TAPI as an example. The TAPI model is a Pruned and Refactored derivative of the Core model. The presentation will highlight how the TAPI model drives tooling to generate Yang and via Yang to generate Swagger definitions etc.
- Some of the general principles and patterns, such as the Component-System and recursive abstraction patterns, will be exposed during the presentation.

Key areas

- 5G – No new capabilities required. All capabilities required already appear for other existing technologies from a management/control perspective
  - E.g., slicing is not new from a management-control perspective
- Access Networks – a network is a network
- Automation and Orchestration – MCC and the Control models
- Containerization – Software model and ConstraintDomain
- Disaggregation – Natural aspect of the model
- Implementation – Tooling approach to generation of Yang etc. from models
- Standardization – Direct via ONF and in collaboration with ITU-T, TMF and MEF
- Virtualization – All functions are naturally virtual/emergent, strong separation of concerns of functional from physical (i.e., something that can be measured with a ruler) and support for distinct, related views of networks ensures support for virtualization